Coding example: Banking

About This Test

The purpose of this code test is to show us your skills in:

- Problem solving
- Code structure and OOP concepts
- Reliability and Testing

Please keep these aspects in mind as you develop your solution. Also, your chosen implementation doesn't necessarily have to be the best you can think of, but one that you can implement in the allocated time.

Instructions

- The time for completing this test is 3 (three) hours
- The language used should be Java, unless previously agreed otherwise
- The build, execution and testing should be maven based
- The resulting code should be directly executable
- All code should be tested
- All needed scripts for compiling, testing, deploying and executing should be included, accompanied by Readme file explaining their usage

Important information: This test is time constrained. It will help us judge your ability to work under suboptimal conditions. For us it will be equally important the quality of resultant code as well as what parts of the project you implemented (and the ones you couldn’t) resulting from constrained time. So before focusing on creating perfect model or perfect conditions checking or another minor part, make sure you will have enough time to cover all needed parts of the solution.

Background

Model in which you gonna operate: Banks have customers. Customers have accounts. Accounts hold money. Banks do transfers between accounts in same bank (intra-bank transfer), or communicate to other banks in order to make inter-bank transfers. Account stores list of all transfers and can be asked for the history of transfers.

There can be two types of transfers:

- Intra-bank transfers, between accounts of the same bank. They don't have commissions, they don't have limits and they always succeed.
- Inter-bank transfers, between accounts in different banks. They have 5€ commissions, they have a limit of 1000€ per transfer. And they have a 30% chance of failure. Keep in mind that banks are independent of each other.

Part 1

Define a set of data structures to accurately reflect banks, accounts and transfers. **Make sure that new type of transfers can be added to the bank with minimal effort.**

Part 2

Create an initial system consisting of two banks BankA and BankB. Jose has an account in the Bank1. Antonio and Maria have accounts in the BankB. Jose and Antonio both own Maria 20,000€ each. Help Antonio and Jose pay their debts by developing a transfer agent (TA).

TA is a program that COMMUNICATES with the banks. When the TA receives an order to transfer money from account A to account B, it issues transfers to the banks considering commissions, transfer limits and possibility of transfer failures.

Questions:

Please also give us your answer to the following questions:

- How would you improve your solution?
- How would you adapt your solution if transfers are not instantaneous?